Grottke, Michael

Working Paper

Modelling structural coverage and the number of failure occurrences with non-homogeneous Markov chains

Diskussionspapiere // Friedrich-Alexander-Universität Erlangen-Nürnberg, Lehrstuhl für Statistik und Ökonometrie, No. 41/2001

Provided in cooperation with:
Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU)

Friedrich-Alexander-Universität Erlangen-Nürnberg
Wirtschafts- und Sozialwissenschaftliche Fakultät

Discussion Paper
41 / 2001

Modelling Structural Coverage and the Number of Failure Occurrences with Non-homogeneous Markov Chains

Michael Grottke

Lehrstuhl für Statistik und Ökonometrie
Lehrstuhl für Statistik und empirische Wirtschaftsforschung
Lange Gasse 20 · D-90403 Nürnberg

Modelling Structural Coverage and the Number of Failure Occurrences with Non-homogeneous Markov Chains

Michael Grottke
Lehrstuhl für Statistik und Ökonometrie
Universität Erlangen–Nürnberg
Lange Gasse 20
D–90403 Nürnberg
Germany
E-Mail: Michael.Grottke@wiso.uni-erlangen.de

October 8, 2001

Abstract

Most software reliability growth models specify the expected number of failures experienced as a function of testing effort or calendar time. However, there are approaches to model the development of intermediate factors driving failure occurrences. This paper starts out with presenting a model framework consisting of four consecutive relationships. It is shown that a differential equation representing this framework is a generalization of several finite failures category models.

The relationships between the number of test cases executed and expected structural coverage, and between expected structural coverage and the expected number of failure occurrences are then explored further.

A non-homogeneous Markov model allowing for partial redundancy in sampling code constructs is developed. The model bridges the gap between setups related to operational testing and systematic testing, respectively. Two extensions of the model considering the development of the number of failure occurrences are discussed.

The paper concludes with showing that the extended models fit into the structure of the differential equation presented at the beginning, which permits further interpretation.

1 A model framework

Most software reliability growth models specify the expected number of failure occurrences μ as a function of some measure of time. The authors of one of the earliest models, Jelinski and Moranda [3], for example, use calendar time t^* as the exogenous variable. However, they also state that an implicit assumption of their model is that the intensity of testing is constant over time. For cases in which this proposition does not hold they formulate a refined model specifying two consecutive mappings: the one from calendar time to testing effort t, and the one from testing effort to the expected number of failure occurrences.

More recently, there have been efforts to identify intermediate factors driving the number of failure occurrences. Gokhale et al. [2] unify several finite failures category models of the Poisson type by considering structural coverage c of the application under test as such a driving factor. Structural coverage can be generically defined as the number of code constructs (for example, statements, blocks or paths) exercised by testing divided by the total number of code constructs in the application under test.

Rivers and Vouk [6, 7] specify the relationship between either test case coverage b - the percentage of planned test cases which have been executed - or structural coverage and the number of failure occurrences. They note that the number of failures experienced is not necessarily proportional to the coverage attained. Their solution is the introduction of an additional factor g, which they call testing efficiency.

Piwowarski et al. [5] derive block coverage, a specific structural coverage, as a random variable C *stochastically* depending on the number of test cases executed i and discuss the expected structural coverage $\kappa(i) = E(C(i))$ and the mean value function $\mu(i)$, assuming that $\mu(i)$ and $\kappa(i)$ are proportional. If the total number of planned test cases, i_t, is known, both functions can be formulated with test case coverage $b = \frac{i}{i_t}$ being the exogenous variable.

Combining all these approaches, four consecutive relationships are identified:

I. The allocation of testing effort t to calendar time t^*

II. The development of test case coverage b as a function of cumulative testing effort

III. The expected coverage of code constructs $\kappa = E(C)$ attained through test case coverage

IV. The relationship between expected structural coverage and the expected number of failures experienced μ

Moreover, a model framework integrating the different concepts can be formulated as the following differential equation:

$$\begin{aligned} \frac{d\mu(\kappa(b(t(t^*))))}{dt^*} &= \frac{dt(t^*)}{dt^*}\frac{d\kappa(b(t(t^*)))/dt(t^*)}{1-\kappa(b(t(t^*)))} \times \\ &\quad g(\kappa(b(t(t^*))))[\nu_d - \mu(\kappa(b(t(t^*))))] \end{aligned} \tag{1}$$

Under the condition that the testing efficiency g is constant, ν_d represents the expected number of failures to be experienced until full structural coverage has been attained. In a model assuming that as soon as a failure has occurred the fault which caused it is

removed instantaneously and perfectly, ν_d is equal to the expected number of detectable faults present in the software at the beginning of testing. If the detection probability is the same time-invariant constant d for all faults, then $\nu_d = \nu \cdot d$. For software reliability models of the binomial type, ν stands for the fixed but unknown number of initial faults, u_0. For models of the Poisson type, it represents the expected value N of the initial number of faults, which is assumed to follow a Poisson distribution.

It is straightforward to show that this framework includes the models mentioned above as well as additional ones:

- The Goel-Okumoto model [1] starts out with a measure of testing effort. Therefore, the relationship I is not in its scope. Moreover, the relationships II, III and IV are not specified explicitly. It is only known that the product $\frac{d\kappa(t)/dt}{1-\kappa(t)} g(\kappa(t))$ is a constant value, say ϕ. A sufficient but not necessary condition is that both the hazard rate of the execution of a certain code construct and the testing efficiency take constant values. In the Goel-Okumoto model, the number of initial faults is a random variable following a Poisson distribution, and the detection probability is assumed to be equal to one. Consequently, in the specific differential equation ν_d is replaced by N. This yields

$$\frac{d\mu(t)}{dt} = \phi[N - \mu(t)]. \tag{2}$$

- Using the same propositions as the Goel-Okumoto model but assuming that the initial number of faults is fixed at the unknown value u_0 leads to the differential equation of the Jelinski-Moranda model [3],

$$\frac{d\mu(t)}{dt} = \phi[u_0 - \mu(t)].$$

- If relationship I is specified by some function $t(t^*) = W(t^*)$ and the product $\frac{d\kappa(t)/dt}{1-\kappa(t)} g(\kappa(t))$ is assumed to be a constant ϕ, then for $\nu = N$ and $d = 1$ the Goel-Okumoto model with a time-varying testing effort [8] is obtained:

$$\frac{d\mu(t^*)}{dt^*} = \frac{dW(t^*)}{dt^*} \phi[N - \mu(t^*)] \tag{3}$$

  Note that one could as well transform the time scale by $t = W(t^*)$ and consider $\mu(t) = \mu(W(t^*))$ and $\frac{d\mu(t)}{dt} = \frac{d\mu(W(t^*))}{dW(t^*)}$, because the differential equation

$$\frac{d\mu(W(t^*))}{dW(t^*)} = \phi[N - \mu(W(t^*))]$$

  can be written as

$$\frac{\frac{d\mu(W(t^*))}{dt^*}}{\frac{dW(t^*)}{dt^*}} = \frac{d\mu(W(t^*))}{dt^*} \frac{dt^*}{dW(t^*)} = \phi[N - \mu(W(t^*))],$$

  which is equivalent to equation (3).

- Again, substituting u_0 for N in equation (3) yields a model of the binomial type, namely the refined Jelinski-Moranda model taking into account a time-varying testing effort [3]:

$$\frac{d\mu(t^*)}{dt^*} = \frac{dW(t^*)}{dt^*}\phi[u_0 - \mu(t^*)]$$

- The Enhanced Non-homogeneous Poisson Process (ENHPP) framework by Gokhale et al. [2] starts out with a measure of testing effort t. Structural coverage is supposed to be a deterministic function of this measure. This means that $\kappa(t) = E(C(t)) = c(t)$. Under the assumptions that the faults are distributed uniformly over the code constructs, that a fault causes a failure with constant probability d the first time that the construct at which it is located is exercised and that faults are removed instantaneously and perfectly as soon as they have caused a failure, the number of failure occurrences increases proportionally to structural coverage. Moreover, the per-fault hazard rate is equal to $\frac{dc(t)/dt}{1-c(t)}$, an expression which can be interpreted as the hazard rate of the execution of a certain code construct if $\lim_{t\to\infty} c(t) = 1$. This implies that the testing efficiency g is identical to one. Therefore, the differential equation of this model framework for non-homogeneous Poisson processes is

$$\frac{d\mu(t)}{dt} = \frac{dc(t)/dt}{1 - c(t)}[Nd - \mu(t)].$$

  The ENHPP framework itself includes a number of finite failures category models. For example, for a deterministic coverage function with a constant hazard rate ϕ and $d = 1$ the Goel-Okumoto model (2) is obtained. A coverage hazard function of $\frac{\phi^2 t}{1+\phi t}$ leads to Ohba's inflection S-shaped model [4] if $d = 1$. Remarkable is the new interpretation given by the ENHPP framework: The shape of the mean value function is determined by the shape of structural coverage as a function of testing effort.

- Rivers and Vouk [6, 7] treat test case coverage and structural coverage as equivalent and exchangeable, implying their numerical identity $c = c(b) = b$. The respective coverage measure is assumed to be an exogenous variable, i.e., the relationships I and II are not in the scope of the model. Moreover, as mentioned above, Rivers and Vouk introduce a factor g, whose value depends on current coverage, for taking into account that coverage growth and the development of the number of failures experienced are not necessarily proportional to each other. In the original formulation, the number of initial faults in the software is not specified to follow a Poisson distribution, but rather treated as an unknown, fixed value. The fault detection probability is constant at one. Therefore, the differential equation of the Rivers-Vouk model is

$$\frac{d\mu(b)}{db} = \frac{g(b)}{1 - b}[u_0 - \mu(b)]$$

  when starting out with test case coverage and

$$\frac{d\mu(c)}{dc} = \frac{g(c)}{1 - c}[u_0 - \mu(c)].$$

when data on structural coverage is to be used.

These forms represent a meta model which needs further specification of the testing efficiency function. For example, setting $g(b) = \alpha(1 - b)$ in the first case yields the so-called linear testing efficiency model, $\frac{d\mu(b)}{db} = \alpha[u_0 - \mu(b)]$. Obviously, its shape is similar to the one of the Jelinski-Moranda model; the main difference is that the exogenous variable is test case coverage, not testing effort.

- In the model by Piwowarski et al. [5] structural coverage growth is not deterministic like in the ENHPP framework, but a stochastic function of the number of test cases executed, i, or test case coverage, $b = \frac{i}{i_t}$. As a consequence, the mean value function is determined by the time-varying function of *expected* structural coverage, $\kappa(b) = E(C(b))$. The expected number of failure occurrences is assumed to be proportional to expected structural coverage (i.e., $g(b) \equiv 1$), and all the u_0 faults in the software are thought to be detectable. Consequently, the differential equation of the model is given by

$$\frac{d\mu(b)}{db} = \frac{d\kappa(b)/db}{1 - \kappa(b)}[u_0 - \mu(b)].$$

Figure 1 summarizes the structures of these models.

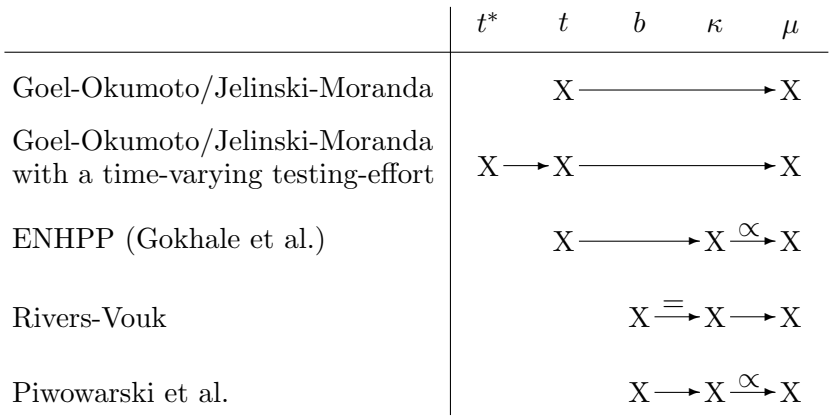

Figure 1: Relationships (implicitly) specified by different models; assumed equalities (=) and proportionalities ($\propto$) are indicated

The framework introduced here does not only unify a number of models in one generalizing equation. It also helps to structure assumptions according to the different consecutive relationships and to determine those propositions that have not been stated explicitly. Focussing on the intermediate relationships may help practitioners to better understand the shape of specific software reliability growth models. Moreover, the suggested forms of such relationships may be discussed with regard to their assumptions about the real world (e.g., the testing strategy used) and - if necessary - improved.

In the remaining sections of this paper, the relationships between the number of test cases executed and structural coverage, and between structural coverage and the number of failure occurrences are explored further.

2 Modelling structural coverage

2.1 Piwowarski et al. versus Rivers and Vouk

In both the approach by Piwowarski et al. and the Rivers-Vouk model the relationship between test case coverage and structural coverage growth is contained.

Piwowarski et al. [5] base their model on the following assumptions:

1. The program under test consists of G code constructs.
2. Per test case, p of these constructs are sensitized on average.
3. The p code constructs are always chosen from the entire population. The fact that a construct has already been sensitized does not diminish its chances of being sensitized in future.

This setup resembles operational testing with a homogeneous operational profile, in which all non-overlapping and equally-sized operations have the same occurrence probability. Piwowarski et al. show that the shape of expected structural coverage attained as a function of the number of test cases executed, $\kappa(i)$, is an exponential one, similar to the one of the Jelinski-Moranda model and the Goel-Okumoto model; i.e.,

$$\kappa(i) = 1 - \exp\left(-\frac{p}{G} \cdot i\right). \tag{4}$$

In figure 2, this equation is depicted as the solid line.

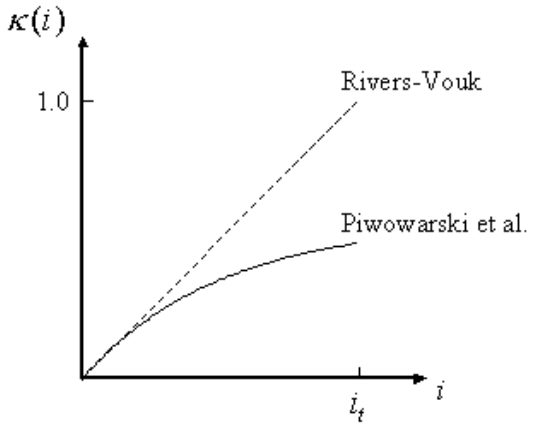

Figure 2: Structural coverage growth in the model by Piwowarski et al. and in the Rivers-Vouk model

On the contrary, the model by Rivers and Vouk [6, 7] is explicitly designed for systematic testing. An objective of this testing regime is to test as much of the application under test as possible. This implies that redundant executions of code constructs have to be avoided. While under the condition of equally-sized test cases the first two assumptions of the model of Piwowarski et al. also apply, assumption three has to be replaced by the central proposition of the Rivers-Vouk model:

3. Code constructs which have already been exercised are not tested a second time.

Clearly, the gain in structural coverage is then $\frac{p}{G}$ for each of the equally-sized test cases, and

$$\kappa(i) = c(i) = \frac{p}{G} \cdot i. \tag{5}$$

This relationship is represented by the dotted straight line in figure 2.

The two models mark extremes. Neither will testers re-execute the same portions of code over and over again during operational testing nor will testers following a systematic approach be able to perfectly avoid redundant executions of code constructs. A model allowing for partial redundancy could capture more realistic situations and bridge the gap between models for operational testing and those for systematic testing. In the following sections, variations of such a model are developed.

2.2 A Markov model for structural coverage

In order to consider partial redundancy in code construct execution, it seems useful to define three different states code constructs may take: "untested" (U), "already tested with the possibility of being tested again in future" (T) and "tested and eliminated from further consideration" (E). The assumptions of the model are as follows:

1. The program under test consists of G code constructs. At the beginning of testing, all these constructs are in state U.

2. Per test case, p constructs are sensitized on average.

3. The p constructs are randomly chosen from those constructs residing in state U or in state T at the beginning of the test case execution.

4. A constant fraction r ($0 \leq r \leq 1$) of those constructs exercised by a test case change to (or stay in) state T and may be tested again in future. The other constructs are eliminated and take state E.

This setup is a generalization of the one by Piwowarski et al. and the one by Rivers and Vouk, because its assumptions 3 and 4 comprise both variations of proposition 3 in the two models.

It can be formulated as a discrete-time Markov chain, in which transitions of the code constructs from one of the three states to another occur at the "atomic" test case executions. A code construct in state U may either remain in this state or switch to one of the other states. Since code constructs already tested before cannot become untested again, they can only stay in the current state or get eliminated. State E is an absorbing (or "closed") one, i.e., constructs eliminated never leave this state. The structure of the resulting Markov chain is depicted in figure 3. $\pi_{AB,i}$ denotes the transition probability from state A to state B during execution of the i^{th} test case. Since the transition probabilities are not time-invariant, the Markov chain is non-homogeneous.

A construct can only change its state if it is sensitized by the test case. According to assumption 2 the probability z_i of being selected by the i^{th} test case is

$$z_i = \frac{p}{E(G_{U,i-1}) + E(G_{T,i-1})}$$

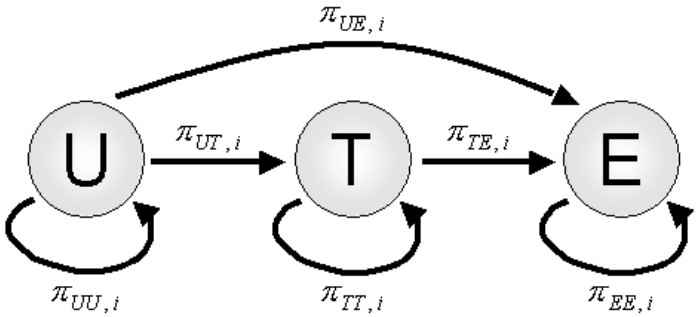

Figure 3: Transition graph of the basic Markov model

for all constructs residing in state U or in state T at the beginning of the i^{th} test case execution, with $E(G_{A,i-1})$ representing the expected number of constructs in state A after execution of the $(i-1)^{th}$ test case.

Since the fraction $(1-r)$ of the code constructs sensitized is eliminated, the transition probabilities $\pi_{UE,i}$ and $\pi_{TE,i}$ are both $z_i(1-r)$. A construct changes from state U to state T if it is exercised but not eliminated; therefore, $\pi_{UT,i} = z_i r$. The probability of remaining in state U, $\pi_{UU,i}$, is equal to $1 - \pi_{UT,i} - \pi_{UE,i} = 1 - z_i$, which is exactly the probability of not being selected. Likewise, $\pi_{TT,i} = 1 - \pi_{TE,i} = 1 - z_i(1-r)$, which can also be calculated as the probability of not being exercised plus the probability of being sensitized but not eliminated. The transition probability $\pi_{EE,i}$ is equal to one, because state E is absorbing.

With the help of the transition matrix of this basic model, $\mathbf{P_i} = \{\pi_{AB,i}\}_{A,B\in\{U,T,E\}}$, the expected number of code constructs in the three states can be expressed as

$$\begin{pmatrix} E(G_{U,i}) \\ E(G_{T,i}) \\ E(G_{E,i}) \end{pmatrix} = \begin{pmatrix} 1-z_i & 0 & 0 \\ z_i r & 1-z_i(1-r) & 0 \\ z_i(1-r) & z_i(1-r) & 1 \end{pmatrix} \begin{pmatrix} E(G_{U,i-1}) \\ E(G_{T,i-1}) \\ E(G_{E,i-1}) \end{pmatrix}. \tag{6}$$

We are interested in the expected number of code constructs covered by the first i test cases, $E(Q_i)$, which is the sum of $E(G_{T,i})$ and $E(G_{E,i})$:

$$\begin{aligned} E(Q_i) &= \begin{pmatrix} 0 & 1 & 1 \end{pmatrix} \begin{pmatrix} E(G_{U,i}) \\ E(G_{T,i}) \\ E(G_{E,i}) \end{pmatrix} = \\ &= z_i E(G_{U,i-1}) + E(G_{T,i-1}) + E(G_{E,i-1}). \end{aligned} \tag{7}$$

Equations (6) and (7) could be used together with the initial conditions $E(G_{U,0}) = G_{U,0} = G$, $E(G_{T,0}) = G_{T,0} = 0$ and $E(G_{E,0}) = G_{E,0} = 0$, which follow from assumption 1, for iteratively calculating the expected number of constructs in the different states and the number of code constructs sensitized after each test case. However, it is also possible to derive a (continuously approximated) closed form expression for the expected number of code constructs exercised, or for expected structural coverage.
Since $E(G_{T,i-1}) + E(G_{E,i-1}) = E(Q_{i-1})$, subtracting $E(G_{T,i-1}) + E(G_{E,i-1})$ from both sides of equation (7) yields

$$E(\Delta Q_i) = z_i E(G_{U,i-1}). \tag{8}$$

It is now necessary to express both z_i and $E(G_{U,i-1})$ in terms of $E(Q_i)$. Obviously, the expected number of constructs not tested during the first $(i-1)$ test cases is exactly the

number of all constructs minus the expected number of constructs that have already been sensitized: $E(G_{U,i-1}) = G - E(Q_{i-1})$.

As for the denominator of z_i, $E(G_{U,i-1}) + E(G_{T,i-1})$, it is equal to $G - E(G_{E,i-1})$. Since a constant fraction $(1-r)$ of the p constructs executed by a test case is eliminated - regardless of the state these constructs were in before -, the change in the number of eliminated constructs is always $p(1-r)$. Therefore, $E(G_{E,i-1})$ is nothing but the deterministic value $G_{E,i-1} = p(1-r)(i-1)$.

Consequently, equation (8) can be written as

$$E(\Delta Q_i) = p \cdot \frac{G - E(Q_{i-1})}{G - p(1-r)(i-1)} \tag{9}$$

or, dividing both sides by the total number of code constructs G, it can be formulated in terms of structural coverage:

$$E(\Delta C_i) = \frac{p}{G} \cdot \frac{1 - E(C_{i-1})}{1 - \frac{p}{G}(1-r)(i-1)} \tag{10}$$

It is possible to continuously approximate difference equation (10) by assuming that each test case can be split in any number of smaller test cases, i.e., by letting $\Delta i \to di \to 0$:

$$\frac{d\kappa(i)}{di} = \frac{p}{G} \cdot \frac{1 - \kappa(i)}{1 - \frac{p}{G}(1-r)i}.$$

Integrating this differential equation finally yields

$$\kappa(i) = 1 - \left(1 - \frac{p}{G}(1-r)i\right)^{\frac{1}{1-r}} \tag{11}$$

for $0 \leq r < 1$ and $\frac{p}{G}(1-r)i < 1$, and

$$\kappa(i) = 1 - \exp\left(-\frac{p}{G} \cdot i\right) \tag{12}$$

for $r = 1$.

These equations confirm that this Markov model is a generalization of the two approaches described in section 2.1: Equation (12) obtained for $r = 1$, i.e., if constructs are never eliminated, is identical to equation (4) derived by Piwowarski et al. Assuming perfect avoidance of redundancy ($r = 0$), on the other hand, leads to the linear relationship (5), which is connected to the setup proposed by Rivers and Vouk.

3 Markov models for the number of failure occurrences

3.1 Common assumptions

The basic Markov model derived in section 2.2 can be extended to include relationship IV of the model framework. This requires to distinguish between correct and faulty code constructs. Like in the existing models explicitly specifying this relationship (e.g. the Rivers-Vouk model), an implicit assumption of this extended Markov model is that at most one fault can be located at a code construct and cause a failure. For code metrics partitioning the software in a large number of code constructs, this proposition does not seem to be very restrictive.

In order to facilitate the distinction between correct and faulty code constructs, each of the states U, T and E is split into two: code constructs can be untested and faulty (UF), untested and correct (UC), tested and faulty (TF), tested and correct (TC), eliminated and faulty (EF) or eliminated and correct (EC).

This entails the necessity to restate the model assumptions:

1. The program under test consists of G code constructs. At the beginning of testing, u_0 of these constructs are in state UF (i.e., they are untested and faulty); the remaining $(G - u_0)$ constructs are in state UC.

2. Per test case, p constructs are sensitized on average.

3. The p constructs are randomly chosen from those constructs residing in one of the states UF, UC, TF or TC at the beginning of the test case execution.

4. A constant fraction r $(0 \leq r \leq 1)$ of those constructs exercised by a test case may be tested again in future. If such a construct is faulty after the test case execution, it changes to (or stays in) state TF; if it is correct after the test case execution, it moves to (or remains in) state TC. The other constructs are eliminated and take state EF or state EC, respectively.

These rephrased assumptions only lay the foundation for the model extension, they do not change the way in which structural coverage grows.

As for the kind of development of the number of failures experienced, i.e. the structure of transitions from the states UF and TF to the states TC and EC, two variations are specified and discussed in the following subsections.

3.2 Fault detection at the first code construct execution only

As we have seen in section 1, several models explicitly specifying relationship IV assume that a failure may occur when a code construct at which a fault is located is exercised *for the first time*. A failure is then either caused with certainty (like in the approach by Piwowarski et al.) or - more generally - only with a certain fault detection probability d (like in the ENHPP framework).

Within the Markov model a proposition corresponding to this setup is as follows:

5. When a code construct at which a fault is located is exercised for the first time, the fault causes a failure with activation probability s ($0 \leq s \leq 1$). The fault is then removed instantaneously and perfectly. If no failure occurs during the first execution of the code construct, then the fault will not be detected until the end testing.

Figure 4 represents the resulting transition graph. The superscript e_I of the transition probabilities indicates that they are linked to the first extended model. The "transitions" within one state have been omitted in order to maintain a clear diagram.

Since the structure in which code constructs are sampled has not been changed, the transition probabilities between the "correct" states are similar to the corresponding ones in the basic model. For example, $\pi^{e_I}_{UCTC,i}$ is still the probability of being selected multiplied by the replacement rate r. However, when calculating the selection probability the expected number of constructs in one of the states UC, TC, UF and TF has to be considered now, i.e.,

$$z_i^{e_I} = \frac{p}{E(G_{UC,i-1}) + E(G_{TC,i-1}) + E(G_{UF,i-1}) + E(G_{TF,i-1})}.$$

Likewise, $\pi^{e_I}_{UCEC,i} = \pi^{e_I}_{TCEC,i} = z_i^{e_I}(1-r)$.

As for the transitions from and between "faulty" states, the fault activation probability s comes into play. Since an untested, faulty construct switches to state TC if it is exercised but not eliminated and if a failure occurs, $\pi^{e_I}_{UFTC,i} = z_i^{e_I} rs$. Similarly, $\pi^{e_I}_{UFTF,i} = z_i^{e_I} r(1-s)$ and $\pi^{e_I}_{UFEF,i} = z_i^{e_I}(1-r)(1-s)$.

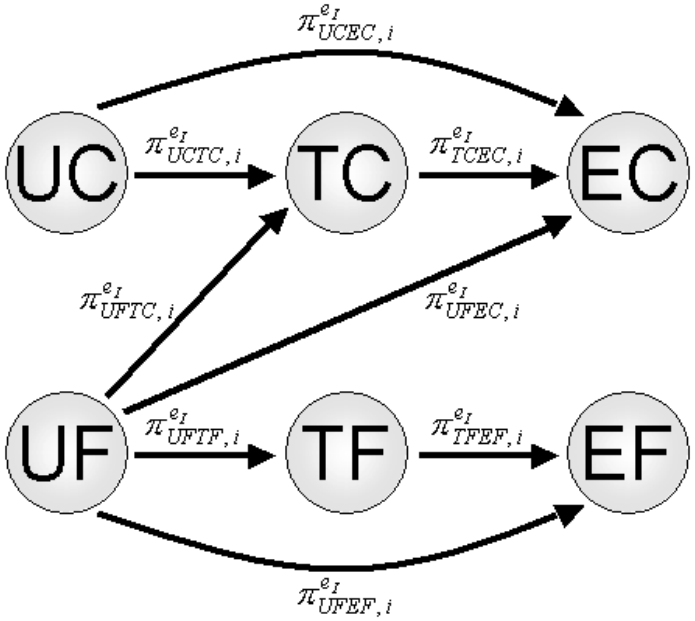

Figure 4: Transition graph of the first extended Markov model

A faulty construct that has already been tested before can only remain in state TF or move to state EF, because according to assumption 5 the fault will never be detected and corrected. Therefore, $\pi^{e_I}_{TFEF,i} = z_i^{e_I}(1-r)$.

This leads to the transition equation of the first extended model

$$\begin{pmatrix} E(G_{UC,i}) \\ E(G_{UF,i}) \\ E(G_{TC,i}) \\ E(G_{TF,i}) \\ E(G_{EC,i}) \\ E(G_{EF,i}) \end{pmatrix} = \mathbf{P_i^{e_I}} \begin{pmatrix} E(G_{UC,i-1}) \\ E(G_{UF,i-1}) \\ E(G_{TC,i-1}) \\ E(G_{TF,i-1}) \\ E(G_{EC,i-1}) \\ E(G_{EF,i-1}) \end{pmatrix}$$

with the non-stationary transition matrix

$$\mathbf{P_i^{e_I}} = \begin{pmatrix} 1-z_i^{e_I} & 0 & 0 & 0 & 0 & 0 \\ 0 & 1-z_i^{e_I} & 0 & 0 & 0 & 0 \\ z_i^{e_I} r & z_i^{e_I} rs & 1-z_i^{e_I}(1-r) & 0 & 0 & 0 \\ 0 & z_i^{e_I} r(1-s) & 0 & 1-z_i^{e_I}(1-r) & 0 & 0 \\ z_i^{e_I}(1-r) & z_i^{e_I}(1-r)s & z_i^{e_I}(1-r) & 0 & 1 & 0 \\ 0 & z_i^{e_I}(1-r)(1-s) & 0 & z_i^{e_I}(1-r) & 0 & 1 \end{pmatrix}.$$

The expected number of code constructs exercised by the first i test cases can be calculated as follows:

$$\begin{aligned} E(Q_i) &= \begin{pmatrix} 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} E(G_{UC,i}) \\ E(G_{UF,i}) \\ E(G_{TC,i}) \\ E(G_{TF,i}) \\ E(G_{EC,i}) \\ E(G_{EF,i}) \end{pmatrix} = \\ &= z_i^{e_I} \left[E(G_{UC,i-1}) + E(G_{UF,i-1}) \right] + \sum_{A \in \{TC,TF,EC,EF\}} E(G_{A,i-1}) \end{aligned}$$

Since $\sum_{A \in \{TC,TF,EC,EF\}} E(G_{A,i-1}) = E(Q_{i-1})$,

$$E(\Delta Q_i) = z_i^{e_I} \left[E(G_{UC,i-1}) + E(G_{UF,i-1}) \right].$$

Like in the basic model, the denominator of the selection probability $z_i^{e_I}$ is equal to $G - p(1-r)(i-1)$. Moreover, the constructs in the states UC and UF are exactly those which have not been tested before; therefore, $E(G_{UC,i-1}) + E(G_{UF,i-1}) = G - E(Q_{i-1})$.

The difference equation derived for the number of code constructs covered is consequently

$$E(\Delta Q_i) = p \cdot \frac{G - E(Q_{i-1})}{G - p(1-r)(i-1)},$$

exactly the same as equation (9). This means that the continuously approximated functions for expected structural coverage derived with regard to the basic model - i.e., equations (11) and (12) - also apply to the first extended model.

This result is not really surprising, because when restating the model assumptions care has been given not to change the way in which coverage grows.

The expected number of faults still remaining in the software after execution of the i^{th} test case is given by $E(G_{UF,i}) + E(G_{TF,i}) + E(G_{EF,i})$. Therefore, $E(M_i)$, the expected number of failure occurrences for the first i test cases, is

$$E(M_i) = u_0 - \begin{pmatrix} 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix} \begin{pmatrix} E(G_{UC,i}) \\ E(G_{UF,i}) \\ E(G_{TC,i}) \\ E(G_{TF,i}) \\ E(G_{EC,i}) \\ E(G_{EF,i}) \end{pmatrix} =$$
$$= u_0 - (1 - z_i^{e_I} s) E(G_{UF,i-1}) - E(G_{TF,i-1}) - E(G_{EF,i-1}).$$

Using the relationship $E(M_{i-1}) = u_0 - E(G_{UF,i-1}) - E(G_{TF,i-1}) - E(G_{EF,i-1})$, it follows:

$$\Delta E(M_i) = z_i^{e_I} s E(G_{UF,i-1}) \tag{13}$$

Quite intuitively, this equation states the reason for growth in the number of failure occurrences: faulty constructs whose execution may cause failures are sensitized, and the fault is activated.

Since untested faulty and untested correct constructs have the same chance of being selected, the proportion of $E(G_{UF,i-1})$ at all expectedly untested constructs is always the same, $\frac{u_0}{G}$:

$$\begin{aligned} E(G_{UF,i-1}) &= \frac{u_0}{G} \left[E(G_{UC,i-1}) + E(G_{UF,i-1}) \right] = \\ &= \frac{u_0}{G} \left[G - E(Q_{i-1}) \right] = \\ &= u_0 \left[1 - E(C_{i-1}) \right] \end{aligned} \tag{14}$$

For $r < 1$, plugging equation (14) into equation (13) and letting Δi approach zero yields

$$\begin{aligned} d\mu(i) &= \frac{p}{G - p(1-r)i} s u_0 \left[1 - \kappa(i) \right] di = \\ &= \frac{p s u_0}{G} \left[1 - \frac{p}{G}(1-r)i \right]^{\frac{r}{1-r}} di. \end{aligned}$$

The continuously approximated function of the expected number of failure occurrences $\mu(i)$ is obtained by integrating this equation:

$$\begin{aligned} \mu(i) &= \frac{p s u_0}{G} \int_0^i \left[1 - \frac{p}{G}(1-r)x \right]^{\frac{r}{1-r}} dx = \\ &= u_0 s \left[1 - \left(1 - \frac{p}{G}(1-r)i \right)^{\frac{1}{1-r}} \right] = \\ &= u_0 s \kappa(i) \end{aligned}$$

For $r = 1$, the continuous approximation of $\Delta E(M_i)$ is

$$\begin{aligned} d\mu(i) &= \frac{p}{G} s u_0 \left[1 - \kappa(i) \right] di = \\ &= \frac{p s u_0}{G} \exp\left(-\frac{p}{G} i \right) di, \end{aligned}$$

and the mean value function is given by

$$\begin{aligned}\mu(i) &= \frac{psu_0}{G}\int_0^i \exp\left(-\frac{p}{G}x\right)dx = \\ &= u_0 s\left[1-\exp\left(-\frac{p}{G}i\right)\right] = \\ &= u_0 s\kappa(i). \end{aligned} \tag{15}$$

In the Markov model, the proposition that only the first execution of a faulty construct may produce a failure leads to a proportional relationship between structural coverage and the number of failure occurrences, just like in the ENHPP framework and in the approach by Piwowarski et al. The constant of proportionality is the expected number of detectable faults, $u_0 s$.

3.3 Fault detection at repeated code construct executions

If a fault is not necessarily detected during the first execution of the construct at which it is located, proportionality between structural coverage and the number of failure occurrences does not seem realistic. Rather, repeated testing of a construct which is still faulty may finally reveal the fault. Therefore, the occurrence of failures should be possible even if no additional structural coverage is attained. With this respect, redundancy may also have positive effects. (Indeed, these are the effects researchers advocating operational testing rely on.)

To account for that, assumption 5 of the first extended model is replaced by the following one:

5. When a code construct at which a fault is located is exercised for the first time or repeatedly, the fault causes a failure with constant activation probability s ($0 \leq s \leq 1$). The fault is then removed instantaneously and perfectly.

This means that in the transition graph of the second extended model (cf. figure 5) additional transitions from state TF to the states TC and EC have to be considered. The transition probabilities are equal to the corresponding ones starting from state UF, i.e., $\pi^{e_{II}}_{TFTC,i} = \pi^{e_{II}}_{UFTC,i} = z_i^{e_{II}} rs$ and $\pi^{e_{II}}_{TFEC,i} = \pi^{e_{II}}_{UFEC,i} = z_i^{e_{II}}(1-r)s$. At the same time, the transition probabilities $\pi^{e_{II}}_{TFTF,i}$ and $\pi^{e_{II}}_{TFEF,i}$ are diminished. The superscript e_{II} indicates the expressions related to the second extended model. However, note that

$$z_i^{e_{II}} = \frac{p}{E(G_{UC,i-1}) + E(G_{TC,i-1}) + E(G_{UF,i-1}) + E(G_{TF,i-1})}$$

is exactly the same as the selection probability in the first extended model, $z_i^{e_I}$.

The transition matrix $\mathbf{P_i^{e_{II}}}$ of this model variation is

$$\begin{pmatrix} 1-z_i^{e_{II}} & 0 & 0 & 0 & 0 & 0 \\ 0 & 1-z_i^{e_{II}} & 0 & 0 & 0 & 0 \\ z_i^{e_{II}} r & z_i^{e_{II}} rs & 1-z_i^{e_{II}}(1-r) & z_i^{e_{II}} rs & 0 & 0 \\ 0 & z_i^{e_{II}} r(1-s) & 0 & 1-z_i^{e_{II}}(1-r(1-s)) & 0 & 0 \\ z_i^{e_{II}}(1-r) & z_i^{e_{II}}(1-r)s & z_i^{e_{II}}(1-r) & z_i^{e_{II}}(1-r)s & 1 & 0 \\ 0 & z_i^{e_{II}}(1-r)(1-s) & 0 & z_i^{e_{II}}(1-r)(1-s) & 0 & 1 \end{pmatrix}.$$

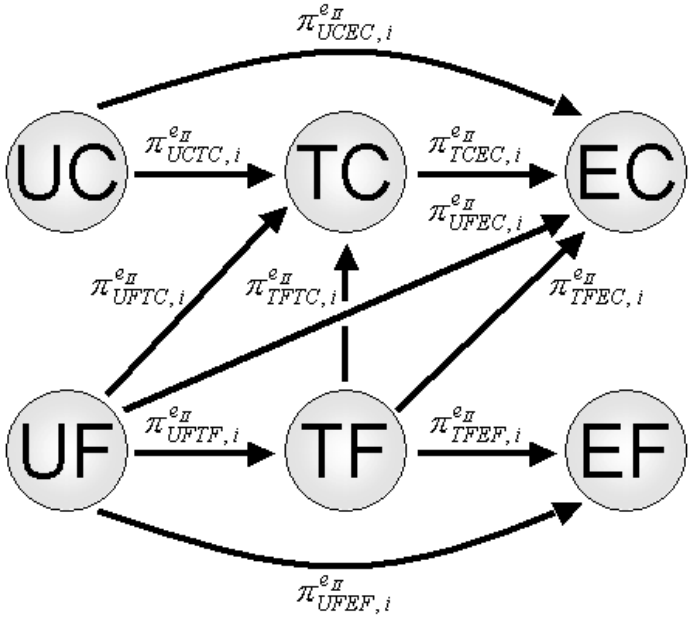

Figure 5: Transition graph of the second extended Markov model

Since no changes have been made with regard to the way in which code constructs are covered, equations (11) and - for $r = 1$ - (12) still hold true.

As for the expected number of failure occurrences after execution of the i^{th} test case, calculating

$$E(M_i) = u_0 - \begin{pmatrix} 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix} \begin{pmatrix} E(G_{UC,i}) \\ E(G_{UF,i}) \\ E(G_{TC,i}) \\ E(G_{TF,i}) \\ E(G_{EC,i}) \\ E(G_{EF,i}) \end{pmatrix}$$

leads to

$$\begin{aligned} \Delta E(M_i) &= z_i^{e_{II}} s \left[E(G_{UF,i-1}) + E(G_{TF,i-1}) \right] = \\ &= z_i^{e_{II}} s E\left(G_{\{UF,TF\},i-1} \right), \end{aligned} \tag{16}$$

where $E\left(G_{\{UF,TF\},i-1}\right)$ denotes $[E(G_{UF,i-1}) + E(G_{TF,i-1})]$. The difference between this equation and the corresponding one for the first extended model lies in the contribution of the constructs in state TF to the number of failures experienced.

If infinitely small changes in i are allowed, equation (16) can be continuously approximated by

$$d\mu(i) = \frac{p}{G - p(1-r)i} s \gamma_{\{UF,TF\}}(i) di. \tag{17}$$

$\gamma_{\{UF,TF\}}$, the continuous approximation of $E\left(G_{\{UF,TF\}}\right)$, has to be determined in order to derive $\mu(i)$.

Note that no transition is directed to the set {UF, TF}. Therefore, the expected number of code constructs leaving this set during the i^{th} test case can be explained by

the expected number of code constructs taking one of the two states before that test case execution. Specifically,

$$\begin{aligned}\Delta E\left(G_{\{UF,TF\},i}\right) &= -E\left(G_{\{UF,TF\},i-1}\right)\left(\pi^{e_{II}}_{UFTC,i}+\pi^{e_{II}}_{UFEC,i}+\pi^{e_{II}}_{UFEF,i}\right)= \\ &= -E\left(G_{\{UF,TF\},i-1}\right)\left(\pi^{e_{II}}_{TFTC,i}+\pi^{e_{II}}_{TFEC,i}+\pi^{e_{II}}_{TFEF,i}\right)= \\ &= -E\left(G_{\{UF,TF\},i-1}\right) z_i^{e_{II}}\left(1-r+rs\right).\end{aligned}$$

For $\Delta i \to di \to 0$, this difference equation transforms into the differential equation

$$\frac{d\gamma_{\{UF,TF\}}(i)}{\gamma_{\{UF,TF\}}(i)} = -\frac{\frac{p}{G}(1-r+rs)}{1-\frac{p}{G}(1-r)i}di,$$

whose integration under the condition $r < 1$ yields

$$\gamma_{\{UF,TF\}}(i) = u_0\left(1-\frac{p}{G}(1-r)i\right)^{\frac{1-r+rs}{1-r}}. \tag{18}$$

Integrating equation (17) with equation (18) plugged in finally results in

$$\mu(i) = u_0\frac{s}{1-r+rs}\left[1-\left(1-\frac{p}{G}(1-r)i\right)^{\frac{1-r+rs}{1-r}}\right]. \tag{19}$$

For $r = 1$,

$$\gamma_{\{UF,TF\}}(i) = u_0\exp\left(-\frac{p}{G}si\right),$$

and

$$\mu(i) = u_0\left[1-\exp\left(-\frac{p}{G}si\right)\right].$$

The expected number of failure occurrences at full structural coverage is $u_0\frac{s}{1-r+rs}$, which lies between $u_0 s$ in case of no redundancy and u_0 for $r = 1$. If a fault is not detected with certainty through execution of the respective defective construct, one can only expect that testing will finally lead to the removal of all faults if code constructs are always replaced.

It can now be proved that in this second extended model the relative number of failures experienced per percentage of structural coverage gained increases. For $0 \le r < 1$,

$$\frac{d\left(\frac{\mu(i)}{\kappa(i)}\right)}{di} = \frac{u_0 s\frac{p}{G}\left(1-\kappa(i)\right)^{rs}}{\left(\kappa(i)\right)^2}\left[1-\frac{rs-r}{1-r+rs}\left(1-\kappa(i)\right)-\frac{1}{1-r+rs}\left(1-\kappa(i)\right)^{r(1-s)}\right].$$

Both the fraction and the expression in brackets are always positive for $\kappa(i) \in (0,1)$ if s is larger than zero.

For $r = 1$,

$$\begin{aligned}\frac{d\left(\frac{\mu(i)}{\kappa(i)}\right)}{di} &= \frac{u_0\frac{p}{G}}{\left(\kappa(i)\right)^2}\times \\ &\quad\left[s\exp\left(-\frac{p}{G}si\right)+(1-s)\exp\left(-\frac{p}{G}(1+s)i\right)-\exp\left(-\frac{p}{G}i\right)\right].\end{aligned} \tag{20}$$

There is no doubt about the positive sign of the fraction. The expression in brackets can be visualized in a graph showing the exponential function $\exp\left(-\frac{p}{G}x\right)$, where x is the exogenous variable (cf. figure 6).

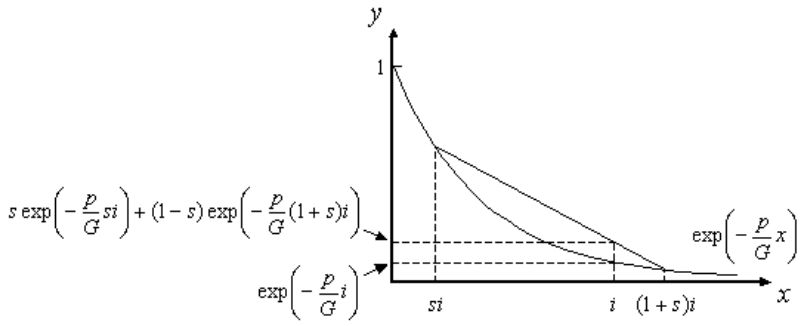

Figure 6: Graphical interpretation of the bracketed expression in equation (20)

For any values $i > 0$ and $0 < s < 1$, the secant between the points $\left(si, \exp\left(-\frac{p}{G}si\right)\right)$ and $\left((1+s)i, \exp\left(-\frac{p}{G}(1+s)i\right)\right)$ can be drawn. Then $s\exp\left(-\frac{p}{G}si\right)+(1-s)\exp\left(-\frac{p}{G}(1+s)i\right)$ is exactly the y-value of that point on the secant for which x is equal to i. Since the concave side of the exponential function points upward, this value is always larger than the y-value of the corresponding point on the curve, $\exp\left(-\frac{p}{G}i\right)$. This means that the bracketed expression and the derivative of $\frac{\mu(i)}{\kappa(i)}$ with respect to i for $r = 1$ are positive for $0 < s < 1$.

Regardless of the extent of redundancy, as long as the detection of faults is neither certain nor impossible the number of failure occurrences per percentage of structural coverage gained increases as testing proceeds.

4 The Markov models and the model framework

It remains to be examined whether the functions derived for the two extended Markov models fit into the differential equation (1) presented at the beginning of this article, or whether this equation has to be generalized further.

In sections 2.2 and 3, expected structural coverage κ and the expected number of failure occurrences μ have been formulated in terms of the absolute number of test cases executed. They can easily be restated as functions of test case coverage $b = \frac{i}{i_t}$, with i_t denoting the total number of test cases planned.

For both extended models, expected structural coverage is the same:

$$\kappa(b) = \begin{cases} 1 - \left(1 - \frac{p}{G}(1-r)i_t b\right)^{\frac{1}{1-r}} & \text{if } 0 \le r < 1 \\ 1 - \exp\left(-\frac{p}{G}i_t b\right) & \text{if } r = 1 \end{cases}$$

In the first extended model, in which failures may only occur at the first execution of a faulty construct, the expected number of failures experienced in terms of test case coverage is

$$\mu(b) = \begin{cases} u_0 s\left[1 - \left(1 - \frac{p}{G}(1-r)i_t b\right)^{\frac{1}{1-r}}\right] & \text{if } 0 \le r < 1 \\ u_0 s\left[1 - \exp\left(-\frac{p}{G}i_t b\right)\right] & \text{if } r = 1 \end{cases}.$$

It can easily be shown that for both cases the following differential equation holds:

$$\frac{d\mu(b)}{db} = \frac{d\kappa(b)/db}{1-\kappa(b)}\left[u_0 s - \mu(b)\right]$$

This equation is a special case of equation (1). This formulation confirms two properties of the model already discussed in section 3.2: Firstly, $\nu_d = u_0 s$ is the number of faults expected to be detected when full structural coverage has been attained. Moreover, since the factor g present in equation (1) is identical to one, the expected number of failure occurrences develops proportionally to structural coverage.

The mean value function of the second extended model in terms of test case coverage b is:

$$\mu(b) = \begin{cases} u_0 \frac{s}{1-r+rs} \left[1 - \left(1 - \frac{p}{G}(1-r) i_t b\right)^{\frac{1-r+rs}{1-r}}\right] & \text{if } 0 \leq r < 1 \\ u_0 \left[1 - \exp\left(-\frac{p}{G} s i_t b\right)\right] & \text{if } r = 1 \end{cases}$$

Again, the differential equation being valid for both cases of this model,

$$\frac{d\mu(b)}{db} = \frac{d\kappa(b)/db}{1-\kappa(b)}(1-r+rs)\left[u_0 \frac{s}{1-r+rs} - \mu(b)\right], \tag{21}$$

is a special case of equation (1). As already noted before in section 3.3, the expected number of detectable faults ν_d is $u_0 \frac{s}{1-r+rs}$. This means that the detection probability d is given by $\frac{s}{1-r+rs}$. Equation (21) additionally reveals that the second extended model implies a constant testing efficiency $g = 1 - r + rs$, which is smaller than one for $r, s < 1$. Interestingly, the value of the testing efficiency is equal to the denominator of the detection probability. Therefore, it looks like that there exists a connection between testing efficiency and the number of detectable faults not discussed in literature so far.

5 Conclusions

The model framework presented in this paper seems to be useful for systematically examining existing models as well as deriving new ones. Generalizing the relationship between test case coverage and structural coverage as specified in the model by Piwowarski et al. and in the Rivers-Vouk model, respectively, the basic Markov model bridges the gap between models for data collected during operational testing and those intended for data collected during systematic testing. Moreover, the model can be extended in various ways in order to consider the expected number of failure occurrences. While the first variation discussed implies a constant testing efficiency of one (and therefore proportionality between structural coverage and the number of failures experienced), the relationship between structural coverage and the number of failure occurrences derived for the second variation involves a constant testing efficiency smaller than one if the activation probability s and the degree of redundancy r are both less than one-hundred per cent.

Instead of constant values for r and s, time-varying probabilities could be examined as well. Even for complicated assumptions, the expected structural coverage and the mean value function in terms of the number of test cases executed can easily be calculated iteratively. However, the derivation of closed-form expressions may not be possible then.

Acknowledgements

The research described in this paper was done in the course of the project PETS. This project is supported by the European Community in the framework of the specific program for research, technological development and demonstration on a user friendly society (1998-2002), the "IST Program". The author is solely responsible for this paper. It does not represent the opinion of the Community.

References

[1] Goel, A. L.; Okumoto, K.: *Time-Dependent Error-Detection Rate Model for Software Reliability and Other Performance Measures*, IEEE Trans. Reliability 28 (1979), pp. 206 - 211

[2] Gokhale, S. S.; Philip, T.; Marinos, P. N.; Trivedi, K. S.: *Unification of Finite Failure Non-Homogeneous Poisson Process Models through Test Coverage*, Proc. Seventh International Symposium on Software Reliability Engineering, White Plains, 1996, pp. 299 - 307

[3] Jelinski, Z.; Moranda, P.: *Software Reliability Research*, in: Freiberger, W. (ed.): *Statistical Computer Performance Evaluation*, New York, 1972, pp. 465 - 484

[4] Ohba, M.: *Software reliability analysis models*, IBM Journal of Research and Development 28 (1984), pp. 428 - 443

[5] Piwowarski, P.; Ohba, M.; Caruso, J.: *Coverage Measurement Experience During Function Test*, Proc. Fifteenth International IEEE Conference on Software Engineering (ICSE), 1993, pp. 287 - 301

[6] Rivers, A. T.: *Modeling Software Reliability During Non-Operational Testing*, Ph.D. thesis, North Carolina State University, 1998

[7] Rivers, A. T.; Vouk, M. A.: *Resource-Constrained Non-Operational Testing of Software*, Proc. Ninth International Symposium on Software Reliability Engineering, Paderborn, 1998, pp. 154 - 163

[8] Yamada, S.; Ohtera, H.; Narihisa, H.: *Software Reliability Growth Models with Testing-Effort*, IEEE Trans. Reliability 35 (1986), pp. 19 - 23